JCL Interview Questions:

1. WHAT IS THE DIFFERENCE BETWEEN SB37,SD37,SE37.

SB37- Abend occurs when the specified primary & secondary is not enough.

SD37- When secondary space is not specified.

SE37- Same like SB37, primary & secondary is not enough, but occurs in case of PDS.

2. HOW TO EXECUTE STEP2,STEP1&STEP3 WHEN STEP1,STEP2,STEP3 ARE IN ORDER.

```
//MAC678AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=MAC678.RANJITHA.PDS(PS1),DISP=SHR
//SYSUT2 DD SYSOUT=(*,INTRDR)
//SYSIN DD *
        TYPE=EDIT, INCLUDE=(STEP2,STEP1,STEP3)
/*
//
```

3. HOW TO ALLOCATE DYANAMIC DS IN JCL.

Mod in Disp parameter is used to allocate dynamic ds if they don't exist, they just append data at the end

4. HOW MANY MAX STEPS WE CAN USE IN JOB.

Max - 255 steps

5. I WANT TO AVOID THE POSITIONAL PARAMETERS HOW TO CODE IT?

Bypassing any positional parameters has to be informed to the system by using ',' then proceeding with keyword parameters.

For eg: //JOBNAME JOB ',' NOTIFY=&SYSUID

6. IS THERE ANY POSSIBBILITY FOR SUBMITTING MORE THAN ONE JOB IN A SINGLE JCL.IF YES, BASED ON WHICH THING THEY WILL EXECUTE.

```
//MAC678AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD SYSOUT=(*.INTRDR)
//SYSUT2 DD DSN=MAC678.RANJITHA.PDS(PS1),DISP=SHR
//
```

PS1 will have another job. When we are using the INTRDR-INTERNAL READER,that will execute a job from another job.

7. SUPPOSE THERE ARE 10 STEPS, OUT OF 10 STEPS. I WANT TO EXECUTE ONLY THE 7TH STEP.

In the job card, RESTART=STEP7, COND=(0,LE). Then the 7th step will be executed, and the remaining steps will be checked with condition. When the condition is true, the remaining steps will be bypassed.

8. HOW CAN U FIND OUT WHETHER THE FILE IS EMPTY OR NOT.

We need to use a simple control card which will check the input file rec count.

```
// JOB001AB JOB NOTIFY=&SYSUID

//STEP1 EXEC PGM=IDCAMS

//SYSPRINT DD SYSOUT=*

//INPUT DD DSN=JOB001.RANJITHA.INFILE, DISP=SHR

//SYSIN DD *

        PRINT INFILE(INPUT) COUNT(1)

/*
```

If the RC=00 then the file is not empty. If RC=04, THEN it is empty.

9. ABOUT RESTART PARAMETER IN JOBCARD.

Restart parameter in the job card is used to specify which step has to be executed. If I have 10 steps and if the job abends at step3 then I don have to start my execution from step1. If I specify the step3, it will bypass the previous steps and will execute the step which I mentioned in the job card.

10. IN GDG I HAVE CREATED 1ST VERSION IN 1ST STEP,I WANT TO USE THE SAME VERSION IN THE 2ND STEP.SO IN WHICH WAY WE CAN USE.

```
//JOB001AB JOB NOTIFY=&SYSUID

//STEP1 EXEC PGM=IEFBR14

//DD1 DD DSN=JOB001.RANJITHA.GDG(+1),

//            DISP=(NEW,CATLG,DELETE),

//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),

//            SPACE=(TRK,(12,10),RLSE)

//SYSOUT DD SYSOUT=*

//SYSPRINT DD SYSOUT=*

//STEP2 EXEC PGM=IEBGENER

//DD1 DD DSN=JOB001.RANJITHA.GDG(0),DISP=SHR
```

```
//DD2 DD DSN=JOB001.RANJITHA.GDG(+1),DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

11. HOW CAN U SAVE THE LOG DETAILS OF A JOB, AFTER THE EXECUTION OF A JOB.

Type the command **XDC** before the name in the spool and hit enter, then it will give the dataset where the output is saved for future verification by the client.

Step1:Use the command XDC before the job name is enter

Step1:Give the dataset name (LRECL=133,RECFM=VB)

NOTE: Whenever we go for XDC, the dataset has to be created with LRECL=133, RECFM=VB.

12. HOW CAN U RESTART A PROC FROM JCL.

In JOBCARD RESTART=PROCNAME.STEPNAME(name of the step in PROC)

13. OVERRIDING DD STATEMENT IN PROC.

To override a DD statement in proc, the sytax should be coded as

//stepname-in-proc.DD name of proc step DD parameter=value.

14. HOW CAN U PASS THE DATA FROM JCL TO COBOL PROGRAMME.

There are two ways,

1. **PARM:**

   Used to pass data from JCL to COBOL. Maximum length is 100 CHAR.

2. **SYSIN DD *:**
   Used to pass the data more than 100 bytes.

JCL

```
//JOBCARD
//S1 EXEC PGM=IEFBR14,PARM="4-1-2012"
```

COBOL

```
LINKAGE SECTION
01 LS-PARM
  49 LS-PARM-LEN PIC S9(4) COMP.
  49 LS-PARM-TEXT PIC X(100).
PROCEDURE DIVISION UISING LS-PARM
  DISPLAY LS-PARM-TEXT.
```

15. WHAT IS THE LENGTH OF PARM PARAMETER.

100 bytes

16. STEP1

STEP2

; HOW U WILL EXECUTE STEP3 ONLY.

;

STEP5

In the job card, RESTART=STEP3, COND=(0,LE). Then the 3rd step will be executed, and the remaining steps will be checked with condition. When the condition is true, the remaining steps will be bypassed.

17. I WANT TO RUN STEP3 EVEN THE JOB IS GIVING ABEND CODE.

COND=EVEN on STEP3. This will execute the 3rd step. Even the previous step executed successfully or unsuccessfully.

18. WHAT IS GDG.HOW U WILL CREATE GDG.

Generation data group(GDG) is a group of datasets where all the datasets are related to each other functionally or chronologically. The datasets in the GDG are called generations.

CREATION OF GDG:

```
//JOB001AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IDCAMS
//SYSIN DD *
        DEFINE GDG(NAME(JOB001.RANJITHA.GDG)-
                        LIMIT(5)-
                        NOEMPTY/EMPTY-
                        NOSCRATCH/SCRATCH)
/*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

19. HOW DO U CREATE GDG BASE.

```
//JOB001AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IDCAMS
//SYSIN DD *
        DEFINE GDG(NAME(JOB001.RANJITHA.GDG)-
                        LIMIT(5)-
```


i-One Techsolutions

NOEMPTY-

NOSCRATCH)

/*

//SYSOUT DD SYSOUT=*

//SYSPRINT DD SYSOUT=*

//

20. WHAT ARE THE DISP PARAMETERS.

DISP=(STATUS,NORMAL,ABNORMAL)

1. **STATUS:**

A. **NEW:** Dataset does not exist. It will be created in this step.
B. **SHR:** Dataset already exist and can be shared.
C. **OLD:** Dataset already exist but can't be shared.
D. **MOD:** If the dataset does not exist, it is to be created. If already exist, records are to be added to the end of the dataset.

2. **NORMAL TERMINATION:**

A. **CATLG:** The dataset has to be catalogued.
B. **DELETE:** Dataset is no longer required.
C. **PASS:** Dataset is passed to subsequent steps in the same job and each step can use the dataset only once.
D. **KEEP:** Dataset is to be kept on volume.
E. **UNCATLG:** System is to delete the catalog entry.

3. **ABNORMAL TERMINATION.**

A. **CATLG:** The dataset has to be catalogued.
B. **DELETE:** Dataset is no longer required.
C. **KEEP:** Dataset is to be kept on volume.
D. **UNCATLG:** System is to delete the catalog entry.
PASS is not allowed here.

21. CAN WE OVERRIDE THE CONDITION IN JCL..

Using EXEC, we can. COND value of an EXEC statement in the procedure can be added/modified/nullified in the invoking JCL in the following way.

//STEP1 EXEC PROC, COND.STEPNAME-IN-PROC=NEW-VALUE

22. DCB=(LRECL=80,RECFM=FB,BLKSIZE=0) HERE WE CAN SEE THE OUTPUT WHEN UR TRYING TO COPY THE DATA FROM INPUT FILE TO OUTPUT FILE.

The dataset can be created with BLKSIZE=0, and we can use it as a output dataset. But the dataset cannot be readed(i.e. we cant use it as a input).

23. IN DISP=(____,_____,____) AFTER SUCCESSFUL COMPILATION IF THE JOB ABENDS ALSO,I HAVE TO SEE THE OUTPUT IN THE OUTPUT FILE,FOR THAT WHAT WE HAVE TO CODE IN DISP.

DISP=(NEW,CATLG,CATLG)

24. MAX TIME LIMIT IN JCL.

TIME=1440hrs

=248Days

=NO LIMIT

=MAX

To overcome from the time abend S322, code anyone of the above.

25. CAN WE OVERRIDE PROC'S WITHOUT USING REFERBACK & SYMBOLIC PARAMETERS.

Yes, we can as follows. PROC-STEPNAME.DDNAME

EX: //S1.DD1 DD DSN=FSS257.Ranji.PDS,SPACE=(TRK,(1,5,2),RLSE)

26. BY USING THE SORT UTILITY,HOW CAN WE COPY FROM PS FILE TO VSAM FILE.

SORT FIELDS=COPY

By using the sort utility

//JOBCARD

//S1 EXEC PGM=SORT

//SORTIN DD DSN=FSS172.RANJI.PS1,DISP=SHR

//SORTOUT DD DSN=FSS172.RANJI.KSDS,DISP=SHR

//SYSIN DD *

SORT FIELDS=COPY

/*

27. ABOUT JOB CARD

//JOBNAME POSITIONAL-PARAMETERS KEYWORD-PARAMETERS

This is the first statement in JCL.

1. **JOBNAME:**
   It identifies the name of the job. The job is identified in the JES SPOOL using this name. There are some rules to code this name field as follows:
   - They may be 1-8 character long.
   - Characters may be alphanumeric/national(#,@,$)
   - Begins at 3rd column and special characters are not allowed.
2. **POSITIONAL PARAMETERS:**

Its mandatory. The following are the Positional parameters in JOB statement.

a. Accounting information: It specifies the account number or account code to which billing has to be done for executing the programs on the mainframe system.without the account number, the job cannot be submitted. Max 142 characters.
b. Program name: programmer name or program functionally or group can be mentioned. It is used for documentation(max 20 chars).

3. **KEYWORD PARAMETERS :**

Keyword parameters are those, which can appear in any order. They are identified by writing the keyword. The following keyword parameters are important for the JOB statement.
CLASS, PRTY, MSGCLASS, MSGLEVEL, TYPRUN, NOTIFY, RESTART.

CLASS: A CLASS parameter categorizes the JOB based on the factors like CPU consumption time and CPU resource utilization etc.

CLASS parameter has a range of values A-Z, 0-9.
JOBS submitted with in the same class parameter will be queued and jobs executed one after the other.
JOBS submitted with the different class parameter will be in a separate queue and are executed simultaneously with the other class jobs.

PRTY: It specifies the priority to a job execution. 'PRTY' has a range of values '0-15' Highest priority value ,the first preference it given for execution, if priority low then execution is also low.

1. When two jobs are submitted, the job which is submitted first will be executed first.
2. If the jobs are submitted at the same time then based on the CLASS parameter the jobs will be executed.
3. If the jobs are submitted with same CLASS parameter and at the same time then the job is executed based on the 'PRTY' parameter.
4. If jobs are submitted with same CLASS parameter with same priority and at the same time then they are kept under wait and after the wait time the operator cancels the job.

MSGCLASS:

It specifies the output device to which messages are sent. It has the range of values A-Z, 0-9.

MSGCLASS = A → printer
= X → spool
= 0 → tool1
= 1 → tool2

MSGLEVEL:

It specifies the 'type' of messages to send and 'when' those messages have to be send.

MESLEVEL = (Statements, Messages)

Statements: -- Type of messages

0 → only job related messages

1 → all JCL related messages includes PROCS and system messages

2 → all JCL related messages excluding PROCS and system messages

Messages: -- when messages have to be sent.

0 → upon successful execution of JOB
1 → upon successful or unsuccessful execution of JOB

Default: -- MSGLEVEL = (1,1)

NOTIFY:

It specifies the user id to which the notification about the successful or unsuccessful execution of the JOB is sent. Because of the "NOTIFY=&SYSUID" we are getting the 'MAXCC=?' value for that particular 'SYSUID'.

- NOTIFY = &SYSUID will substitute the user-id with which the user has been logged in and so the notifications has send to same user-id.
- If we specify a different user-id (NOTIFY=USER ID) then notifications will be sent to this user-id, but the spool remains with the user-id.

REGION:

It specifies the amount of work space required for a job or step execution.

- REGION must be specified in ODB quantities in Kilo or Mega bytes
- The mega region space can be allocating by specifying REGION= 0K or 0M.

Note: -- If the specified Region space is enough then job will abend with S106.

TIME:

It specifies the maximum time allocated for a job or step execution.

Syntax:-- TIME = (Minutes ,Seconds)

=(10,10)

Note:-- Maximum time can be allocated by using

TIME = 1440 Hrs.

= 248 days

= NOLIMIT (or) MAX

TYPERUN:

It tells the system what needs to be done to the JOB.

1. "TYPERUN=SCAN"
   It checks JCL syntax errors or compile JCL.
2. "TYPERUN=HOLD"
   It checks JCL syntax errors but keeps the JOB on HOLD without executing it. To release the JOB on hold specify the line command 'a' beside the job name.
3. "TYPERUN=COPY"
   It copies the JOB to particular location.

RESTART:

It is used to start the execution of the JOB from a particular step

```
// JOB1 JOB……RESTART=STEP3.
//  STEP1 EXEC PGM=PGM1
   ------------
   -------------
//  STEP2 EXEC PGM=PGM2
//  STEP3 EXEC PGM=PGM3
```

28. HOW CAN PASS THE DATA FROM JCL TO COBOL,WHERE WE CAN DECLARE IN COBOL.

There are two ways,

1. **PARM:**

   Used to pass data from JCL to COBOL. Maximum length is 100 CHAR.

2. **SYSIN DD *:**
   Used to pass the data more than 100 bytes.

29. INSTREAM PROCEDURES & CATALOGUE PROCEDURE.

PROCEDURE: Set of reusable executable JCL statements. There are two procedures.

1. **INSTREAM PROCEDURE:**
   - Present inside the same job stream.
   - The statements between PROC and PEND will be considered as a instream procedure.
   - This procedure can be executed maximum of 15 times within a JOB.
2. **CATALOG PROCEDURE:**
   - will be present in another PDS.
   - this procedure can be called by JCLLIB ORDER
   - can be executed ‘n’ no of time.

30. DIFFERENCE BETWEEN STEPLIB & JOBLIB.

In Mainframe, there are three types of libraries.

1. system library(SYS1.LINKLIB): we don’t need any JOBLIB/STEPLIB to call/execute the program like IEFBR14, IEBGENER,..(UTILITIES) designed by IBM.
2. private/user library: we have to generate a load library where the program can be called and executed.
3. temporary library: dataset will get deleted once it reaches the expiry date.

To execute the programs presented in Private or User library, temporary library, we should use JOBLIB/STEPLIB.

JOBLIB: valid for all the steps in the job.

STEPLIB: valid only for the particular step where it is coded.

If both are coded, then steplib will override the joblib.

ANS:

In mainftame there are three lib

(i) SYSLIB(SYS1.LINKLIB)

(ii) USER/PRIVATE LIB

(iii) TEMPARARY LIB

SYSLIB:The programs which are presented in system library have the load modules these are generate by an IBM

--> To execute the program present in SYSLIB no need to specify the STEPLIB AND JOBLIB

USER/PRIVATELIB: THE PROGRAMS CODED BY A PROGRAMERS ARE CALLED AS USER OR PRIVATE PROGRAMS

--> TO EXECUTE THIS PROGRAMS THE LOAD LIBRARY PRESENTED IN A PRIVATE LIBRARIES

-> tO EXECUTE THIS PROGRAM WE WILL USE STEPLIB AND JOBLIB

TEMPARARYLIB: THESE LIBRARIES ARE CREATED FOR A PARTICULAR DURATION OF A PROJECT

--> tO EXECUTE THIS LIBRARIES WE SHOULD USED STEPLIB AND JOBLIB

31. I HAVE A INPUTFILE1 WHICH CONTAINS 10 RECORDS,I HAVE 100 RECORDS IN INFILE2,BUT MY REQUIREMENT IS I WANT TO DO APPEND IN INFILE1 WITH 100 RECORDS.WHAT DISP ARE ALL THE RESULTS IN GIVEN.

		WILL APPEND OR NOT	WILL OVERRIDE
DISP=	NEW	X	X
	OLD	X	YES
	SHR	X	YES
	MOD	YES	X

DISP=MOD . This will append those 100 records.

32. //step1 exec _ _ _ _

//step2 exec _ _ _ _ WHAT CONDITION WE HAVE TO WRITE SUCH THAT STEP2 WILL NEVER EXECUTE.

COND=(00,LE) on STEP2. This condition is always true. So it will bypass the step, irrespective of the return code of the previous step.

33. I HAVE A FILE WHICH CONTAINS 100 RECORDS BUT.I WANT TO SORT FIRST 40 RECORDS ONLY & AFTER THAT WRITE IN OUTPUT.HOW U CAN CODE.

//JOBNAME JOB NOTIFY=&SYSUID

//STEP1 EXEC PGM=SORT

//SYSPRINT DD SYSOUT=*

//SORTIN DD DSN=JOB001.RANJITHA.PS1,DISP=SHR

//SORTOF01 DD DSN=JOB001.RANJITHA.PS2,DISP=SHR

//SYSIN DD *

SORT FIELDS= (1,6,CH,A)

OUTFIL FILES=01,STARTREC=01,ENDREC=40

/*

34. HOW WILL U GET TO KNOW THE JCL SYNTAX ERRORS WITHOUT EXECUTING JCL.

TYPERUN=SCAN will check for sytax errors without execution.

35. HOW U WILL CODE TIME LIMIT FOR FORTYFIVE SECONDS.

TIME=(0,45)/ TIME=(,45)

36. DISP=(NEW,PASS,DELETE) WHAT IS THIS EXPLAIN.

A new dataset is to be created temporarily after successful execution. If the job terminates abnormally, delete the dataset.

37. WHAT IS USE OF RLSE IN SPACE PARAMETER.

It uses unused space of any dataset.

38. WHAT IS THE DIFFERENCE BETWEEN INSTREAM PROCEDURE & CATALOG PROCEDURE.HOW U WILL CALL CATALOG PROCEDURE.

INSTREAM PROCEDURE	CATALOG PROCEDURE

1. Procedure will be in the same job stream.
2. Can be called only for 15 times.
3. Instream proc. begins with PROC and ends with PEND.
4. Will be called as follows
//STEP1 EXEC PROCNAME
	1. Procedure will be located in different pds. Not inside the same job stream.
	2. Can be called 'n' no. of times.
	3. Begins with PROC.
	4. Will be called as follows
	// JCLLIB ORDER=LOC.OF.DS
	//STEP EXEC PROC-NAME

39. WHAT ARE SYMBOLIC PARAMETERS.WHERE U CAN USE IN YOUR PROGRAMME.

A symbolic is a PROC placeholder. The value for the symbolic is supplied when the PROC is invoked. (&symbol=value). If the value is not provided during invoke, then the default value coded in the PROC definition would be used for substitution.

Ex: if you want to override UNIT parameter value of all the DD statements, defeine this as symbolic parameter in PROC.

CATALOG PROC: PROC1

//PROC1 PROC, &UNIT=SYSDA =>Sets default value of UNIT AS SYSDA.

//S1 EXEC PGM=TEST1

//DD1 DD UNIT=&UNIT

//DD2 DD UNIT=&UNIT

// STEP1 EXEC PROC,UNIT=TEMPDA will set &UNIT as TEMPDA for this run of procedure.

We can use symbolic overrides in EXEC and DD statements**.**

40. WHAT IS GDG.WHAT IS THE PURPOSE OF GDG.

Generation data group(GDG) is a group of datasets where all the datasets are related to each other functionally or chronologically. The datasets in the GDG are called generations.

GDG dataset are reffered in JCL using GDG base and relative number. So the same JCL can be used again and again without changing the dataset name and this is biggest advantage of GDG. A GDG base has pointer to all its generations. When you want to read all the transactions done till today, you can easily do it by reading the GDG base if is available. Otherwise you have to concatenate all the transaction files before reading.

GDG-(Generation Data Group)

1) It is used for taking backups of data

2) For generating reports

41. I HAVE A JOB WHICH CONTAINS 3 STEPS.

//JOB CARD

//S1 EXEC PGM=----

//DD1 DD DSN= A.B.C(+1)

//S2 EXEC PGM=---

//DD2 DD DSN=

//S3 EXEC PGM=---

//DD3 DD DSN=

HERE I HAVE TO USE THE O/P GENERATION OF STEP1 AS INPUT DS IN S2 & S3.CAN I USE.HOW.

Yes. It is possible. The first step creates one generation as (+1). As it is required to use the generation created in step1, it should be referred as (+1) in STEP2 and STEP3, as the (+1) version is not yet prompted as current version. Because, Generations are updated only at the end of the JOB.

So it should be coded as follows,

//JOB CARD

//S1 EXEC PGM=----

//DD1 DD DSN= A.B.C(+1)

//S2 EXEC PGM=---

//DD2 DD DSN= A.B.C(+1)

//DD3 DD DSN=OUTPUT.PS.NAME

//S3 EXEC PGM=---

//DD4 DD DSN= A.B.C(+1)

//DD5 DD DSN=OUTPUT.PS.NAME

42. IF U WANT TO REFER PREVIOUS GENERATION HOW U WILL.

YES WE CAN USE

//DD1 DD DSN=A.B.C.(-1),DISP=SHR

43. WHAT IS TEMPORARY DATASET.HOW WE CAN DEFINE.IF WE HAVE TWO TEMPORARY DATASETS IF WE EXECUTE SECOND STEP DIRECTLY,WHAT IS STATUS OF FIRST STEP.(MEANS IT WILL KEEP SAME OR ANY ABEND O IT WILL DELETE).

Temp dataset created by step availabe until in job,aftter completion job it automaticaly deleted.,it defined by dsn=&&tempfilename., first temprary dataset keep same

44. HOW MANY VERSIONS ARE THERE IN GDG

min --> 1 generation

max -->255 generation

45. WE HAVE TWO INFILES WE WANT TO MERGE IN SORTED ORDER.

INFILE1	**INFILE2**
2	**1**
4	**3**
5	**7**
6	**8**

Use the SORT

```
//FSS045AS JOB NOTIFY=&SYSUID
//*  MERGING OF TWO DATA SETS ****/
//S1 EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SORTIN1 DD DSN=FSS045.RANJITHA.PS300,DISP=SHR
//SORTIN2 DD DSN=FSS045.RANJITHA.PS400,DISP=SHR
//SORTOUT DD DSN=FSS045.RANJITHA.PS500,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSIN DD *
```

MERGE FIELDS=(1,4,CH,A,1,4,CH,A)

/*

46. IF UR JCL CONSISTS BOTH JOBCAT & STEPCAT THEN WHICH ONE IS MORE EFFICIENT/WHICH ONE HAS HIGHER PREFERENCE.

The datasets used in step are first checked in the STEPCAT (ICF or VSAM Catalog) before checking in the system catalog. If there is no STEPCAT in the step and there is a JOBCAT, then the datasets are first searched in JOBCAT before checking in the system catalog.

ANS: STEPCAT OR STEPLIB

47. WHAT IS CONDITIONAL PARAMETER.

It is used to control the execution of the steps.

SYNTAX:

COND=(COMPARISION CODE, RELATIONAL OPERATOR, RC)

If the 'COND' specified in the step is false then the step executes. And if it is true then the step is bypassed.

TRUE → Bypassed

FALSE → Executes

COND=EVEN" it executes the step even if the previous steps executed successfully or unsuccessfully.

COND=ONLY" it executes the step only if the previous step executes unsuccessful.

COND=(00,LE) is always 'TRUE' condition it bypass the step all the time.

COND=(00,GT) is always 'FALSE' condition it executes the step all the time.

48. HOW WE CAN RENAME THE DS'S.IS IT POSSIBLE TO RENAME THE DS'S.

By specifying R, infront of the dataset, the dataset can be renamed.

ANS: By giving R before the ds name

R FSS257.UMA.PS1(ENTER)

NEW name : PS5 (enter)

then it will be automatically renamed.

49. IF I WANT TO SUBMIT A JOB WITH SOME OTHER USER ID IN THAT CASE.WHAT I HAVE TO DO.

SYNTAX for submitting the job with other USER ID:

//JOB0001A JOB NOTIFY=&SYSUID, USER=OTHER-USERID, PASSWORD=PASSWORD-OF-THE-USER

EX: Suppose I want to submit a job. But I don have access to execute a DB2 TABLE. So in that case I can go for the abve syntax with the USER-ID who has access to execute it(DB2 TABLE).

50. WHAT IS CLASS PARAMETER &PRTY.

CLASS: A CLASS parameter categorizes the JOB based on the factors like CPU consumption time and CPU resource utilization etc.

CLASS parameter has a range of values A-Z, 0-9.
JOBS submitted with in the same class parameter will be queued and jobs executed one after the other.
JOBS submitted with the different class parameter will be in a separate queue and are executed simultaneously with the other class jobs.

PRTY: It specifies the priority to a job execution. 'PRTY' has a range of values '0-15' Highest priority value ,the first preference it given for execution, if priority low then execution is also low.

1. When two jobs are submitted, the job which is submitted first will be executed first.
2. If the jobs are submitted at the same time then based on the CLASS parameter the jobs will be executed.
3. If the jobs are submitted with same CLASS parameter and at the same time then the job is executed based on the 'PRTY' parameter.
4. If jobs are submitted with same CLASS parameter with same priority and at the same time then they are kept under wait and after the wait time the operator cancels the job.

51. WHAT IS COND STATEMENT.

It is used to control the execution of the steps.

SYNTAX:

COND=(COMPARISION CODE, RELATIONAL OPERATOR, RC)

If the 'COND' specified in the step is false then the step executes. And if it is true then the step is bypassed.

TRUE → Bypassed

FALSE → Executes

COND=EVEN" it executes the step even if the previous steps executed successfully or unsuccessfully.

COND=ONLY" it executes the step only if the previous step executes unsuccessful.

COND=(00,LE) is always 'TRUE' condition it bypass the step all the time.

COND=(00,GT) is always 'FALSE' condition it executes the step all the time.

52. WHAT IS MSGCLASS &REGION.

MSGCLASS:

It specifies the output device to which messages are sent. It has the range of values A-Z, 0-9.

MSGCLASS = A → printer

= X → spool

= 0 → tool1

= 1 → tool2

REGION:

It specifies the amount of work space required for a job or step execution.

- REGION must be specified in ODB quantities in Kilo or Mega bytes
- The mega region space can be allocating by specifying REGION= 0K or 0M.

Note: -- If the specified Region space is enough then job will abend with S106.

53. WHAT IS JOBLIB & STEPLIB.WHICH IS HAVING HIGHEST PRIORITY.

JOBLIB: It specifies the load library in which all the steps, program load modules are searched.

STEPLIB: It specifies the load library in which a particular steps load module is searched. It has the highest priority.

54. WHAT IS GDG.WHAT ARE PARAMETERS IN GDG.CAN I DELARE LIKE THIS

//SYSIN DD *

DEFINE GDG(NAME( )

LIMIT(5)

NOEMPTY

SCRATCH)

Generation Data Groups or GDGs are a group of data sets which are related to each other chronologically and functionally. These related Data Sets share a unique Data Set Name.
Every GDG data set has a Generation number and Version number assigned to each data set.

PURPOSE:

1. Used to take back-up of data.
2. It automates the process of creating and deleting datasets.
3. We need not change the JCL when we need to create a new file.

EG.

```
//MYJOB   JOB  NOTIFY=&SYSUID
//STEP1   EXEC PGM=IDCAMS
//SYSIN   DD   *
  DEFINE GDG(NAME(MYLIB.LIB.TEST)    -
         LIMIT(10)            -
         NOEMPTY              -
         SCRATCH)
/*
//
```

PARAMETERS:

NAME - This parameter is used to specify the name of the data set that is to be created.
LIMIT - This parameter is used to specify the total number of generations that the GDG may Contain
EMPTY/NOEMPTY - These two parameters are mutually exclusive. EMPTY specifies that all existing generations of the GDG are to be uncataloged whenever the generations of GDG Reached the maximum limit NOEMPTY specifies that only the oldest generation of the GDG is to be uncataloged if the limit is reached
SCRATCH/NOSCRATCH - These two parameters are mutually exclusive. SCRATCH parameter specifies that whenever entry of the GDG is removed from the index, it should be deleted physically and uncataloged. NOSCRATCH parameter specifies that whenever entry of the GDG is removed from the index, it should be uncataloged, not physically deleted.

55. WHAT IS INREC &OUTREC.

INREC: Structure of the records before sorted order.

OUTREC: Structure of the records after sorting..

In input file, I have to fields but in outrec I want only three fields. In that case, inrec will have records before sorted order. Outrec will say records after sorted order with 3 fields.

56. HOW CAN U CONCATNATE TWO FILES.

Using IEBGENER, we can concatenate maximum of 255 PS.

```
ANS:    //JOB NAME JOB NOTIFY=&SYSUID
        //STEP1 EXEC PGM=IEBGENER
        //SYSUT1 DD DSN=FSS045.TEST.PS1,DISP=SHR,
        //              DD DSN=FSS045.TEST.PS2,DISP=SHR,
        //SYSUT2 DD DSN=FSS045.TEST.PS3,DISP=SHR,
        //SYSIN DD *
        /*
```

Using IEBCOPY, we can concatenate maximum of 15 PDS.

```
ANS:    //JOB NAME JOB NOTIFY=&SYSUID
        //STEP1 EXEC PGM=IEBCOPY
        //DD1 DD DSN=FSS045.TEST.PS1,DISP=SHR,
        //DD2 DD DSN=FSS045.TEST.PS2,DISP=SHR,
        //DD3 DD DSN=FSS045.TEST.PS3,DISP=SHR,
        //SYSIN DD *
                COPY INDD=DD1,DD2, OUTDD=DD3
        /*
```

57. HOW DO U MERGE TWO FILES.

```
//FSS045AS JOB NOTIFY=&SYSUID
//*  MERGING OF TWO DATA SETS ****/
//S1 EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SORTIN1 DD DSN=FSS045.RANJITHA.PS300,DISP=SHR
//SORTIN2 DD DSN=FSS045.RANJITHA.PS400,DISP=SHR
//SORTOUT DD DSN=FSS045.RANJITHA.PS500,DISP=SHR
```

```
//SYSOUT DD SYSOUT=*

//SYSIN DD *

   MERGE FIELDS=(1,4,CH,A,1,4,CH,A)

/*
```

58. IF I HAVE PRESENT YEAR GENERATION,PREVIOUS YEAR GENERATION.PREVIOUS PREVIOUS YEAR HOW CAN I RETRIEVE PREVIOUS PREVIOUS YEAR.

By specifying GDG(-2), We can retrieve the previous previous year generation.

59. IF THERE ARE 5 STEPS I WANT TO EXECUTE ONLY 5TH STEP.IN THAT CASE WHAT IS COND PARAMETER U WILL USE.

There are two ways.

1. by specifying RESTART=STEP5 in JOB card
2. using IEBEDIT as follows:

```
//MAC678AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=MAC678.RANJITHA.PDS(PS1),DISP=SHR
//SYSUT2 DD SYSOUT=(*,INTRDR)
//SYSIN DD *
        TYPE=EDIT, INCLUDE=(STEP5)
/*
//
```

60. WHAT IS OUTREC.WHERE WE WILL USE IT.

OUTREC: Structure of the records after sorting..

61. HOW CAN U ELIMINATE DUPLICATES BY USING SORT UTILITY.

```
//FSS045B JOB NOTIFY=&SYSUID
//S1 EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SORTIN DD DSN=FSS045.RANJITHA.PS1,DISP=SHR
//SORTOUT DD DSN=FSS045.RANJITHA.PS2,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSIN DD *
  SORT FIELDS=(1,10,CH,A)
  SUM FIELDS=NONE
/*
```

62. WHAT IS DIFFERENCE BETWEEN SHR &OLD.

DISP=OLD: It will lock the dataset. Other users cant access the dataset, unless or until the lock is released. i.e.explicitly locks the dataset

DISP=SHR: Everyone can share it.

63. HOW CAN I OVERRIDE ONE FILE WITH ANOTHER FILE.

//STEPNAME.DDNAME DD DSN=FSS045.TEST.PS22,DISP=SHR

64. IN JCL I HAVE 4 STEPS I WANT TO EXECUTE ONLY 3RD & 4TH .I DON'T WANT TO USE RESTART.HOW U WILL DO IT.

Using IEBEDIT as follows:

```
//MAC678AB JOB NOTIFY=&SYSUID
//STEP1 EXEC PGM=IEBEDIT
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=MAC678.RANJITHA.PDS(PS1),DISP=SHR
//SYSUT2 DD SYSOUT=(*,INTRDR)
//SYSIN DD *
        TYPE=EDIT, INCLUDE=(STEP3,STEP4)
/*
//
```

65. WHAT IS TIME PARAMETER.WHAT IS MAX TIME LIMIT.

TIME:

It specifies the maximum time allocated for a job or step execution.

Syntax:-- TIME = (Minutes ,Seconds)

=(10,10)

Maximum time can be allocated by using

TIME = 1440 Hrs.

= 248 days

= NOLIMIT (or) MAX

i-One Techsolutions

Cobol Interview Questions:

1. HOW TO SOLVE U4038 ABEND.

TWO REASONS: 1. Mismatching of logical and physical file.

2. STOP RUN is not specified.

2. I HAVE SEQUENTIAL FILE IN WHICH THERE ARE 50 RECORDS,NOW I WANT TO COPY ALL THE 50 RECORDS IN THE REVERSE ORDER INTO A NEW FILE.THE LAST RECORD IN THE ORIGINAL FILE SHOULD BE THE FIRST RECORD IN THE NEW FILE HOW CAN I DO IT.

1. Using FILE-AID.

2. Perform varying WS-I from 50 by -1 until WS-CNT=1

3. HOW CAN I USE THE PARAMETERS DECLARED IN MAINPROGRAMME IN SUB PROGRAMME.

By using the linkage section and using statement.

4. HOW DO U DEFINE A FILE IN COBOL & JCL.

Cobol:

ENVIRONMENT DIVISION, INPUT-OUTPUT SECTION, FILE-CONTROL.

SYNTAX: SELECT INFILE ASSIGN TO DD1.

Jcl:

//DD1 DD DSN=DATASET-NAME,DISP=SHR

5. DIFFERENCE BETWEEN SYSOUT & SYSPRINT.

SYSOUT:

Used after executing the program. We can see the output in sysout.

SYSPRINT:

We will get after compilation. To check the messages related to your program.

6. IF I HAVE 2 FILES,FILE-1 HAVING DATA & FILE-2 IS EMPTY,HOW CAN U FIND THE FILE-2 IS EMPTY BY USING EMPTY.

ANS: By using EMPTY statement also we can decide whether file is empty or not.

7. HOW CAN WE DECLARE DYNAMIC ARRAY IN COBOL.

WORKING-STORAGE SECTION.

05 WS-STUDENT OCCURS N TIMES.

PROCEDURE DIVISION.

ACCEPT N.

8. HOW MANY TYPES OF CALLS ARE THERE.

Two types. 1. Static call

2. Dynamic call

1. STATIC CALL:

- Program name will be specified in main program.

- the main program and the sub-program load modules are linked together at the time of compilation in the static call.

2. DYNAMIC CALL:

- program, name will be specified while executing the program.

- the main program and the sub-program load modules are linked together at the time of execution.

Difference between MAIN & SUB program?

In MAIN program:

- the variables are declared in the workin-storage section.

- stop run will be specified.

In SUB program:

- the variables are declared in Linkage Section.

- GOBACK will be specified.

9. IN CALL PARAMETER THE VARIABLES MUST BE SAME(WHETHER IN TYPE,LENGTH,NAME...).

Data Type should be the same..Variable name & Length can be different.

10. HOW CAN U USE THE PACKED DECIMAL DATA & HOW & WHERE SIGN WILL STORE.WHY WE DECLARE THE COMP-3.

COMP-3 is used to store the packed decimal values. The memory calculation for COMP-3,

S9(N) COMP-3.

If N is EVEN then it will take ((N/2)+1)bytes. If N is ODD, then it will take (N+1)/2 bytes.

NOTE: Last four bytes used to store the sign of the variable.

11. HOW CAN WE PRINT THE COMP-3 VARIABLE IN THE REPORT.HOW CAN WE DECLARE THE INTERMEDIATE VARIABLE & WHAT IS THE USAGE OF THIS INTERMEDIATE VARIABLE.

Using the command HEX ON in the file, COMP-3 can be printed in the report. To display the COMP-3 variable, use another intermediate variable. i.e., 9(4).9(2) ->displays the packed decimal values.

12. IF A FILE IS ALEREADY EXIST WITH SOME LENGTH.I WANT TO EXTENDING THAT FILE LENGTH.

We cant change the DCB parameter.

13. WHAT IS THE CLUSTERING INDEX.

Clustering index is for each an every VSAM file, it will create a clustering file.

14. HOW CAN U PASS THE DATA FROM COBOL TO JCL.

There are two ways.

1. MOVE 4 TO RC.
   After executing the program, will get the Return Code as 4.

2. While using files, will do the operations on the logical file names in the COBOL programs that reflects the files presented in JCL.

15. WHAT IS FILE STATUS 92.

It is a logical error, if the file is opened in INPUT mode, we can read the records in the file. Whenever, the user tries to UPDATE/WRITE the records into the file. This is a logical mismatch.

16. WHAT IS EVALUATE STATEMENT.

It is used to execute the statements based on condition. Replacing of nested-if statements is called an Evaluate statement.

```
SYNTAX:   EVALUATE TRUE/VARIABLE NAME
          WHEN COND-1
          DISPLAY 'COND-1'
          WHEN COND-2
          DISPLAY 'COND-2'
          WHEN OTHERS
          DISPLAY 'OTHERS'
          END-EVALUATE.
```

WHEN statement is used to specify the condition. We can use maximum of 255 WHEN conditions
WHEN OTHER is always a true condition and hence must be specified as last when condition.
THRU or THROUGH is used to specify a range of values.
END-EVALUATE is the mandatory scope terminator.

17. WHAT IS SEARCH ALL ORDER.

There are two types of SEARCH in an array.

1. SEARCH
2. SEARCH-ALL

SEARCH	SEARCH-ALL
1. It is a linear search.	1. It's a binary search.

2. Array elements are searched one after another. 3. Data need not to be in sorted order. 4. SEARCH is efficient than PERFORM. 5. Multiple WHEN condition can be specified. 6. Any relational operator can be used(LE,GT,GE,LE,EQ,NE)	2. Array is splitted into two halves. Last element of the first half and 1st half and 1st element of the 2nd half are compared with desired value. Based on the possibility of the desired value in one half, the other half is ignored. Remaining half is again split to compare until desired value is obtained. 3. Data need to be in sorted order. 4. SEARCH ALL is efficient than SEARCH 5. One WHEN condition can be coded. 6. Only (=) operator can be used.

18. WHAT IS LEVEL WE WILL USE FOR CONDITION.

88 for CONDITIONS.

19. HOW MANY DIVISIONS ARE THERE IN COBOL.

FOUR DIVISIONS.

1. IDENTIFICATION DIVISION
2. ENIRONMENT DIVISION
3. DATA DIVISION
4. PROCEDURE DIVISION

20. WHAT DO U DEFINE IN THE SCREEN SECTION.

TSO Command, Zoom

21. DIFFERENCE BETWEEN STATIC AND DYNAMIC CALL.

1. STATIC CALL:

- Program name will be specified in main program.

- the main program and the sub-program load modules are linked together at the time of compilation in the static call.

- PARM=NODYNAM

2. DYNAMIC CALL:

- program, name will be specified while executing the program.

- the main program and the sub-program load modules are linked together at the time of execution.

-PARM=DYNAM

22. WHAT IS REDEFINES.

Sharing of memory is called REDEFINES.

Syntax:

```
WORKING-STORAGE SECTION.
01 GRP1.
  05 A PIC X(3) VALUE 'IBM'.
```

01 GRP2 REDEFINES GRP1.
05 B PIC X(10).

Note:

1). Redefines is possible for elementary item to elementary item.
2).Redefines is possible for Group item to Group item.
3). Redefines is not possible for Sub-elementary item to Sub-elementary item.

23. IN WHAT SITUTATIONS WE ARE USING REDEFINES IN COBOL.

While validating one information with other, redefines needed.

24. CAN WE REDEFINE COPYBOOK.

No. its not possible.

25. WHAT IS COMPUTATIONAL FIELD.

There are below types of Computational clauses in COBOL .
1).Comp
2).Comp-1
3).Comp-2
4).Comp-3
5).Comp-4
6).Comp-5

1. **Comp:**

Binary representation of data item.

PIC clause can contain S and 9 only.

S9(01) – S9(04) Half word.

S9(05) – S9(09) Full word.

S9(10) - S9(18)Double word.

Most significant bit is ON if the number is negative.

Syntax: 01 WS-A PIC S9(8) USAGE COMP.

2. **Comp-1:**

Single word floating point item. PIC Clause should not be specified. The sign is contained in the first bit of the of the leftmost byte and the exponent is contained in the remaining 7 bits of the first byte. The last 3 bytes contain the mantissa.

Syntax: 01 WS-A USAGE COMP-1.

3. **Comp-2:**

Double word floating-point item. PIC Clause should not be specified. 7 bytes are used for mantissa and hence used for high precision calculation.

Syntax:01 WS-AUSAGE COMP-2

4. **Comp-3:**

Packed Decimal representation. One digit takes half byte. PIC 9 (N) comp-3 data item would require (N + 1)/2 bytes. The sign is stored separately in the rightmost half-byte regardless of whether S is specified in the PICTURE or not.

C – Signed Positive D – Signed Negative F-Unsigned Positive.

Syntax:01 WS-A PIC 9(2)V9(4) USAGE COMP 3.

Formula to calculate the number of bytes: ODD NUMBER= N+1/2

EVEN NUMBER=N/2+1

5. **COMP-4:**
   BINARY INTEGER DATA

NOTE: binary, comp, comp-4 are all the same; values in these kinds of fields may be truncated, based on compiler options and the particular work being done.

6).**COMP-5:** binary integer, but never truncated

26. WHY WE ARE USING COMP.WHAT IS COMP-3.

1. **Comp:**

Binary representation of data item.

PIC clause can contain S and 9 only.

S9(01) – S9(04) Half word.

S9(05) – S9(09) Full word.

S9(10) - S9(18)Double word.

Most significant bit is ON if the number is negative.

Syntax: 01 WS-A PIC S9(8) USAGE COMP.

2. **Comp-3:**

Packed Decimal representation. One digit takes half byte. PIC 9 (N) comp-3 data item would require (N + 1)/2 bytes. The sign is stored separately in the rightmost half-byte regardless of whether S is specified in the PICTURE or not.

C – Signed Positive D – Signed Negative F-Unsigned Positive.

Syntax:01 WS-A PIC 9(2)V9(4) USAGE COMP 3.

27. HOW MUCH MEMORY S9(7) COMP-3 OCCUPY.

4 bytes. i.e., (7+1)/2 as the no is odd.

28. SYNTAX OF SORT IN COBOL.

In COBOL, SORT is used to sort the records on TEMP FILE using INFILE and giving output.

SYNTAX: SORT TEMPFILE ON ASCENDING KEY ENO OF TEMPREC USING INFILE GIVING OUTFILE.

(If we don specify the tempfile then it will take the spool as TEMPFILE)

29. CAN WE USE COPYBOOK IN PROCEDURE DIVISION.

SYNTAX: COPY MEMBERNAME

30. WHAT IS REPLACING COMMAND ASSOCIATED WITH COPYBOOK.

In COBOL, COPYBOOK is used to reduce the number of lines in coding.

Copy Inrec replacing == AC-no == by == AC-no1 ==

31. WHERE U WILL SPECIFY THE DYNAM & NODYNAM IN STATIC & DYNAMIC CALL.

In RUNJCL/RUNTIME, by using the PARM parameter we will pass the values whether it is DYNAM/NODYNAM.

32. 01 A PIC ---

CAN WE REDEFINE

05 B PIC ---

No. LEVEL Number should be same.

33. 01 A PIC

05 B PIC CAN WE REDEFINE D WITH B.

01 C PIC

05 D PIC

No. BCOZ sub-elementary is not possible in redefine

34. WHAT IS SOC7 ERROR.

Invalid data in numeric field (datatype mismatch).

There are two possibilities of getting this abend.

1. while passing, non-numeric value to numeric field using MOVE statement.
2. There are some non-numeric values are presented in file. Those files are declared as numeric.

SOLUTIONS:

Go to SPOOL, select CESDUMP, get the line no. and offset address. If we are unable to get the exact location, then copy the offset address from the CESDUMP, goto SYSPRINT, type L.OFFSET ADDRESS. Then it will give the particular line no. Now GOTO the Main program, type L line no in Command Line, the cursor will be positioned at particular line. Then edit the values.

35. WHAT ARE THE WAYS OF SPLITTING STRINGS.

1. RENAMES

2. REDEFINES

3. REFERENCE MODIFICATION

4. UNSTRING

1. RENAMES:

```
WORKING-STORAGE SECTION.
01 GRP1.
    05 A PIC X(3) VALUE 'IBM'.
    05 B PIC X(3) VALUE 'TCS'.
    05 C PIC X(5) VALUE 'WIPRO'.
    05 GRP2.
        07 D PIC X(10) VALUE 'IT-INDUSTRY'.
66 R RENAMES C THRU D.
PROCEDURE DIVISION.
    DISPLAY GRP1.
    DISPLAY R.
    STOP RUN.
```

2. REDEFINES

```
WORKING-STORAGE SECTION.
01 NAME1 PIC A(10) VALUE 'ARUN KUMAR'.
01 NAME REDEFINES NAME1.
    02 FIRST-NAME  PIC A(5).
    02 LAST-NAME PIC A(5).
PROCEDURE DIVISION.
    DISPLAY FIRST-NAME.
    DISPLAY LAST-NAME.
    STOP RUN.
```

3. **REFERENCE MODIFICATION**

```
WORKING-STORAGE SECTION.
01 A  PIC A(10) VALUE 'ARUN KUMAR'..
01 B  PIC A(5).
01 C PIC A(5).
PROCEDURE DIVISION.
    MOVE A(1:4) TO B.
    MOVE A(5:5) TO B.
    STOP RUN.
```

36. HOW DO U CALCULATE THE TOTAL NO.OF CHARACTERS IN STRING.

SYNTAX

INSPECT VARIABLE-NAME1 TALLYING WS-CNT FOR ALL CHARACTERS.

37. HOW CAN U FIND THE TOTAL NO.OF RECORDS IN FILE.

```
SYNTAX:
PERFORM UNTIL WS-EOF='Y'
READ INFILE
AT END
MOVE 'Y' TO WS-EOF
NOT AT END
ADD+1 TO WS-CNT
END-READ
END-PERFORM.
CLOSE INFILE
CLOSE OUTFILE
DISPLAY WS-CNT.
```

38. IN FILE HOW U WILL COPY ONLY EVEN RECORDS INTO THE OUTPUT FILE.

```
SYNTAX:
PERFORM UNTIL WS-EOF='Y'
READ INFILE
AT END
MOVE 'Y' TO WS-EOF
NOT AT END
ADD+1 TO WS-CNT
IF WS-CNT=2
MOVE IN-REC TO OUT-REC
WRITE OUT-REC
MOVE 0 TO WS-CNT
END-IF
END-READ
END-PERFORM.
CLOSE INFILE
CLOSE OUTFILE
STOP RUN.
```

39. IS IT POSSIBLE TO DELETE RECORDS FROM ESDS.

DELETE is not possible in ESDS. ADDING records at the end is possible but updating is not possible.

40. WHAT IS SSRANGE/NOSSRANGE.

SUBSCRIPT	INDEX
1. It is an occurrence number of an array element is defined as subscript.	1. It is the displacement value of an array element
2. Must be declared a separate variable in working-storage section.	2. It is declared by using INDEXED BY CLAUSE.
3. It is decremented/incremented using arithmetic	3. Incremented/decremented using SET clause.

operation. 4. Access is slow.	4. Access is fast.

SSRANGE-Subscript Range.

NOSSRANGE- No Subscript Range

PARM=SSRANGE for dynamic array.

PARM=NOSSRANGE for static array. It is default.

41. HOW DID U FIND WHETHER IT IS AN STATIC OR DYNAMIC CALL BY SEEING THE PGM.

SYNTAX: CALL 'PGM-NAME' for static.

CALL VARIABLE-NAME for dynamic.

42. WHAT IS DIFFERENCE BETWEEN INDEX&SUBSCRIPT.

SUBSCRIPT	INDEX
1. It is an occurrence number of an array element is defined as subscript. 2. Must be declared a separate variable in working-storage section. 3. It is decremented/incremented using arithmetic operation. 4. Access is slow.	1. It is the displacement value of an array element 2. It is declared by using INDEXED BY CLAUSE. 3. Incremented/decremented using SET clause. 4. Access is fast.

43. COMP-1 & COMP-2 WHICH ONE IS MOST EFFICIENT FOR STORING THE FLOATING POINT VALUE

1. **Comp-1: It is Single precision floating point value. Maximum 4 byte**

Single word floating point item. PIC Clause should not be specified. The sign is contained in the first bit of the of the leftmost byte and the exponent is contained in the remaining 7 bits of the first byte. The last 3 bytes contain the mantissa.

Syntax: 01 WS-A USAGE COMP-1.

2. **Comp-2: It is double precision floating point value. Maximum 8 byte**

Double word floating-point item. PIC Clause should not be specified. 7 bytes are used for mantissa and hence used for high precision calculation.

Syntax:01 WS-AUSAGE COMP-2

44. SEARCH & SEARCH ALL,WHICH SEARCH IS PREFERABLE FOR LARGER TABLES

SEARCH	SEARCH-ALL
1. It is a linear search. 2. Array elements are searched one after another. 3. Data need not to be in sorted order. 4. SEARCH is efficient than PERFORM. 5. Multiple WHEN condition can be specified. 6. Any relational operator can be used(LE,GT,GE,LE,EQ,NE)	1. It's a binary search. 2. Array is splitted into two halves. Last element of the first half and 1st half and 1st element of the 2nd half are compared with desired value. Based on the possibility of the desired value in one half, the other half is ignored. Remaining half is again split to compare until desired value is obtained. 3. Data need to be in sorted order. 4. SEARCH ALL is efficient than SEARCH 5. One WHEN condition can be coded. 6. Only (=) operator can be used.

SEARCH ALL is preferable for larger table.

45. WHAT ARE THE DISADVANTAGES OF VSAM FILES.

- Requires lot of DASD space.

- For KSDS, primary key cannot be changed.

- Performance can be slow because of complexity of INDEX.

- Only resides on DISK.

46. HOW CAN UKNOW THAT WHICH TYPE OF FILE UR USING IN COBOL.

FILE-CONTROL.

SELECT FILENAME1 ASSIGN TO EXTERNAL REFERENCE.

	SEQUENTIAL	->ESDS
ORGANISATION IS	INDEXED	->KSDS
	RELATIVE	->RRDS
	SEQUENTIAL	->ESDS
ACCESS MODE IS	INDEXED	->KSDS
	RELATIVE	->RRDS

RECORD KEY IS DATANAME1.

47. WHAT IS REFERENCE MODIFICATION.

It is used to move only a part of a field to a part of another field.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. ELEM.
DATA DIVISION.
```

```
WORKING-STORAGE SECTION.
02 A  PIC X(13) VALUE 'PRAVEEN KUMAR'.
02 B  PIC X(7).
02 C  PIC X(5).
PROCEDURE DIVISION.
   MOVE A(1:7) TO B.
   MOVE A(9:5) TO C.
   DISPLAY B
   DISPLAY C
   STOP RUN.
```

48. COBOL IS A OBJECT-ORIENTED OR STRUCTURED PROGRAMMING LANGUAGE.

VS-COBOL2 INTERFACE COBOL.

49. WHICH VERSION UR USING.

ENTERPRISE COBOL/VS COBOL II.

V3.1

50. WHAT IS EVALUATE STATEMENT.

It is used to execute the statements based on condition. Replacing of nested-if statements is called an Evaluate statement.

```
SYNTAX:     EVALUATE TRUE/VARIABLE NAME
            WHEN COND-1
            DISPLAY 'COND-1'
            WHEN COND-2
            DISPLAY 'COND-2'
            WHEN OTHERS
            DISPLAY 'OTHERS'
            END-EVALUATE.
```

WHEN statement is used to specify the condition. We can use maximum of 255 WHEN conditions
WHEN OTHER is always a true condition and hence must be specified as last when condition.
THRU or THROUGH is used to specify a range of values.
END-EVALUATE is the mandatory scope terminator.

51. WHAT IS NEXT & CONTINUE STATEMENT.

If the condition is true then CONTINUE statement execute the statements which are after the last Scope terminator.

If the condition is true then NEXT SENTENCE execute the statements which are after the period(.).

52. IN EVALUATE IF THERE IS A SITUTATION LIKE A=1 AND B=2 HOW CAN I USE IT.

```
EVALUATE TRUE

WHEN A=1 AND B=2

        DISPLAY 'B=2'
```

WHEN A=1 AND B=4

DISPLAY ‘B=4’

WHEN OTHER

DISPLAY ‘NOT VALID’

END-EVALUATE.

53. WHAT ARE THE DIFFERENT DATA TYPES IN COBOL.

NUMERIC -18 DIGITS

ALPHABETIC & ALPHA-NUMERIC -32,767 BYTES

54. CAN I USE SAME COPYBOOK FOR BOTH INFILE AND OUTFILE.

SYNTAX TO REPLACE:

COPY MEM1 REPLACING == EMP-ID == BY == EMP-NO ==

55. WHAT IS SORTING IN COBOL?

In Cobol SORT statement used to sort the input files write into output file either in ascending or descending order.

SYNTAX:- SORT WRKFILE ON ASSENDING KEY ENO OF WRREC USING INFILE GIVING OUTFILE

56. WHAT IS THE DECLARATION TO AN ARRAY.WHAT IS THE SYNTAX.

01 I PIC 9 VALUE 1.

01 ARRAY.

05 ARRAY1 PIC X(9) OCCURS 10 TIMES.

05 ARRAY2 PIC X(20) OCCURS 20 TIMES INDEXED BY I.

57. IN WHAT SITUTATION WE WILL GO FOR REDEFINES.

Sharing of memory is called redefine,redefine group to another group ,elementary to elementary.

58. IN UR PROGRAMME HOW U WILL DECLARE COPYBOOK.

ANS: ENVIRONMENT DIVISION.
COPY WORKCOPY

WORKCOPY:
WORKING-STORAGE SECTION.
01 A PIC X(5) VALUE ‘ARTHI’.
01 B PIC X(13) VALUE ‘IBM-MAINFRAME’

59. HOW CAN U RESOLVE SOC-7.

Invalid data in numeric field (datatype mismatch).

There are two possibilities of getting this abend.

1. while passing, non-numeric value to numeric field using MOVE statement.
2. There are some non-numeric values are presented in file. Those files are declared as numeric.

SOLUTIONS:

Go to SPOOL, select CESDUMP, get the line no. and offset address. If we are unable to get the exact location, then copy the offset address from the CESDUMP, goto SYSPRINT, type L.OFFSET ADDRESS. Then it will give the particular line no. Now GOTO the Main program, type L line no in Command Line, the cursor will be positioned at particular line. Then edit the values.

60. WHAT IS SOC-4.

Storage violation error.

REASONS:

- Missing select statement, during compilation time.

- Bad subscript or index.

- Read/write attempt to unopened file.

- Move data to/from unopened file.

- Missing parameters is called sub-programs.

61. WHAT R THE ERROR CODES U KNOW IN COBOL.

SOC7, SOC4, U4O38, SOCB, SOC1

62. WHAT IS NESTED-IF STATEMENT.

When an IF statement contains an IF statement as one of its possible branches, the IF statements are said to be *nested*. Theoretically, there is no limit to the depth of nested IF statements.

However, use nested IF statements sparingly. The logic can be difficult to follow, although explicit scope terminators and indentation help. When a program has to test a variable for more than two values, EVALUATE is probably a better choice.

The following depicts a nested IF statement:

```
IF CONDITION1

        IF CONDITION2

                STATEMENT1

        ELSE

          STATEMENT-2

         END-IF

         STATEMENT-3
```

ELSE

STATEMENT-4

END-IF

63. WHAT IS COPYBOOK.

It is used to store the predefined record structure. basically used to call the sub routines, Abend modules …..etc. After compilation the information present in copybook will be copied to main program.

Syntax: COPY membername

Example:
IDENTIFICATION DIVISION.
PROGRAM-ID. COPY1
ENVIRONMENT DIVISION.
COPY WORKCOPY
PROCEDURE DIVISION.
DISPLAY A.
DISPLAY B.
STOP RUN.
Copybook contains below information.
WORKCOPY:
WORKING-STORAGE SECTION.
01 A PIC X(5) VALUE 'ARTHI'.
01 B PIC X(13) VALUE 'IBM-MAINFRAME'.

63. WHAT IS EXIT IN COBOL.

EXIT does nothing.

64. WHAT IS SEARCH AND SEARCH ALL.

SEARCH	SEARCH-ALL
1. It is a linear search. 2. Array elements are searched one after another. 3. Data need not to be in sorted order. 4. SEARCH is efficient than PERFORM. 5. Multiple WHEN condition can be specified. 6. Any relational operator can be used(LE,GT,GE,LE,EQ,NE)	1. It's a binary search. 2. Array is splitted into two halves. Last element of the first half and 1st half and 1st element of the 2nd half are compared with desired value. Based on the possibility of the desired value in one half, the other half is ignored. Remaining half is again split to compare until desired value is obtained. 3. Data need to be in sorted order. 4. SEARCH ALL is efficient than SEARCH 5. One WHEN condition can be coded. 6. Only (=) operator can be used.

65. IF I USE GOBACK INSTEAD OF STOPRUN WHAT WILL HAPPEN.

STOP RUN is frequently used and mainly coded in main programs(calling program). The control will be returned to OS.

GOBACK is coded in the mainly in the subroutines or called programs. The control will be returned to the calling program. If STOP RUN is coded in the calling program the program will go to an infinite execution. GOBACK can be coded in the programs which is not called by any programs instead of STOP RUN.

66. 01 A X(5).

01 B X(3).

PROCEDURE DIVISION.

MOVE 'ABCDE' TO B.

DISPLAY B.

WHAT IS VALUE OF B.

B=ABC

67. 01 A 9(5).

01 B 9(3).

PROCEDURE DIVISION.

MOVE 12345 TO B.

DISPLAY B.

WHAT IS VALUE OF B.

B=345

68. S9(5) COMP--HOW MANY BYTES IT WILL OCCUPY.

S9(5) COMP-3----HOW MANY BYTES IT WILL OCCUPY.

S9(5) COMP = 4BYTE

S9(5) COMP-3 = 3BYTE

69. 9(5) COMP-----HOW MANY BYTES IT WILL OCCUPY.

9(5) COMP-3---- HOW MANY BYTES IT WILL OCCUPY.

9(5) COMP = 4BYTE

9(5) COMP-3 = 3BYTE

DB2 Interview Questions:

1. I HAVE A DB2 TABLE WITH 1000 ROWS AFTER UPDATING FIRST 110 ROWS,I GOT ABEND SO,HOW CAN I RESTART FROM 111TH ROW.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Restart1.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 ESAL PIC 9(3).
77 WS-CNT PIC 9(3).
EXEC SQL
 INCLUDE SQLCA
END-EXEC.
EXEC SQL
 INCLUDE DCLGEN1
END-EXEC.
PROCEDURE DIVISION.
        EVALUATE SQLCODE
        WHEN 0
        COMPUTE ESAL = ESAL*110/100
       EXEC SQL
      UPDATE RANJITHA1
      SET ESAL = :ESAL
      END-EXEC
      ADD+1 TO WS-CNT
      IF WS-CNT = 10
      EXEC SQL
        COMMIT
     END-EXEC
     MOVE TO WS-CNT
     END-IF
     WHEN 100 CONTINUE
      END-EVALUATE.
      EXEC SQL
       CLOSE
    END-EXEC
    STOP RUN.
```

2. HOW CAN U IDENTIFY,WHETHER IT IS COBOL-DB2 PROGRAMME BY SEEING THE JCL.

```
// STEP1 EXEC PGM=IKJEFT01

        PARM(PARM NAME,PLAN NAME)
```

3. IF WE WANT TO PASS THE DATA FROM JCL TO COBOL,WE USE PARM=' ', IN EXEC STATEMENTS,IF WE NEED TO PASS THE DATA TO COBOL-DB2 PROGRAMME HOW & WHERE WE CAN SPECIFY PARM PARAMETER.

At RUN-TIME, in SYSTSIN card of IKJEFT01 using PARM parameter.

4. TIME STAMP MISMATCH,PRECOMPILATION PROCESS,NULL INDICATOR.

TIMESTAMP MISMATCH, -818, is a RUNTIME ERROR. Whenever there is a mismatch of timestamp between DBRMLIB & LOADLIB. In that case, we have to REBIND the program.

PRE-COMPILATION PROCESS: COBOL-DB2 Program is first feed to DB2 pre-compiler that extracts the DB2 statements into DBRM and replace the source program DB2 statements with COBOL CALL statements. This modified source is passed to COBOL compiler and then link editor to generate load module. During pre-compilation, time stamp token is placed on modified source and DBRM.

NULL INDICATOR: NULL INDICATOR VARIABLE are used to know, if any NULL values are retived from DB2 into COBOL variables.

- If indicator value is <0 then it indicates NULL value is retrived and since COBOL default values based on the DATA TYPE.
- If indicator value is 0 then value is properly retrived.
- If value is >0 the the data retrived is truncated.

5. CURSORS,ISOLATION LEVELS(CS),AT WHAT TYPE OF SITUTATIONS WE WILL GO FOR CS.

To retrieve more than one row randomly from DB2 table, we are going to use the cursors.

STEPS TO USE CURSOR:

There are four steps to use cursor.

1. Declare cursor
2. Open cursor
3. Fetch cursor
4. Close cursor

NOTE:

-811 abend caused because the user trying to retrieve more than one row, without declaring cursor.

If we want to retrieve more than one row without using cursor, then use FETCH in SELECT statement as follows. In that case the rows will be retrieved sequentially.

Syntax: SELECT * FROM TABLE FETCH FIRST N ROWS ONLY

ISOLATION LEVEL:

There are 3 isolation levels in DB2.

1. Cursor stability (CS). –row level locking
2. Repeatable read (RR). –page level locking
3. Uncommitable read (UR) –it wont lock any record.

6. WHAT IS THE PICTURE CLAUSE FOR NULL INDICATOR.

01 EMP-LANDLINE-NO PIC S9(4) COMP.

Whenever we want to pass value to null, that should be declared in WS section. Then only we can pass.

7. GIVE SOME ABENDS IN DB2.

-805: **REASON:** DBRM/PACKAGE is not found in the PLAN.

-911: **REASON:** DEADLOCK. The program is used by some other.

SOLUTION: submit after sometime.

-922: **REASON:** AUTHOURIZATION FAILURE.

SOLUTION: Send mail to DBA to get access for particular tables. Or client will give the access to complet the JOB.

8. HOW CAN U DISPLAY FIRST ROW IN THE TABLE.

Syntax: SELECT * FROM TABLE FETCH FIRST 1 ROWS ONLY

9. TO SELECT THE NULL DATA.

Syntax: SELECT * FROM TABLE WHERE COLNAME=NULL

10. HOW CAN U SELECT SUBSTRING.

SELECT SUBSTRING (FIELDNAME, STARTING POSITION, LENGTH)

11. HOW CAN WE INSERT NULL VALUES TO A PARTICULAR COLOUMN OF A TABLE.

INSERT INTO TABLENAME VALUES ('A001', 'RAHUL', ' ')

12. DIFFERENCE BETWEEN PRIMARY KEY & UNIQUE INDEX.

PRIMARY KEY	UNIQUE INDEX
1. It is not null & unique. (it won't allow duplicate).	1. Will create the unique index. For the primary key field. It is useful to retrieve the data from DB2 table. (without unique index we cant create primary key).

13. WHAT IS DCLGEN.

DCLGEN-DECLARATION GENERATOR.. it is used to generate the COBOL equivalent data types for the DB2 data types.

14. HOW WE WILL DO NULL DATA SELECTION IN COBOL-DB2 PROGRAMME.

Syntax: SELECT * FROM TABLE WHERE COLNAME=NULL

15. WHEN WE WILL DO REBINDING A PROGRAMME.HOW WE WILL DO REBIND.

When SQL statements are not changed but a new index is added or RUNSTATS is run, then it is advisable to do a REBIND for the best possible access path. REBIND PLAN is cheaper than BIND with ACTION(REPLACE).

SYNTAX:

SYSTSIN

REBIND PLAN(PLAN-NAME)

VALIDATE(BIND)

…..

16. PRECOMPILATION OF COBOL-DB2 PROGRAMME.

SQL(Structered Query Language) is the language that is used to retrieve/update/delete DB2 data. SQL statements are embedded into COBOL program within the scope of 'EXEC SQL and END-EXEC.

COBOL-DB2 Program is first feed to DB2 pre-compiler that extracts the DB2 statements into DBRM and replace the source program DB2 statements with COBOL CALL statements. This modified source is passed to COBOL compiler and then link editor to generate load module. During pre-compilation, time stamp token is placed on modified source and DBRM.

17. WHERE DO U DEFINE CURSOR.

Either in working-storage section or procedure division.

18. IN DB2 HOW DO U GET ROWS IN ONE TABLE.

SELECT COUNT(*) FROM TABLENAME

19. WHAT IS NULL INDICATOR.

NULL INDICATOR VARIABLE are used to know, if any NULL values are retived from DB2 into COBOL variables.

- If indicator value is <0 then it indicates NULL value is retrived and since COBOL default values based on the DATA TYPE.
- If indicator value is 0 then value is properly retrived.
- If value is >0 the the data retrived is truncated.

20. WHAT IS -818 SQLCODE.WHEN THIS ERROR WILL COME WHETHER IN COMPILE/RUNTIME.

TIMESTAMP MISMATCH, -818, is a RUNTIME ERROR. Whenever there is a mismatch of timestamp between DBRMLIB & LOADLIB. In that case, we have to REBIND the program.

21. WHAT IS NULL INDENTIFIER.

22. WHAT IS DCLGEN.DID U USED NULL IDENTIFIER IN DCLGEN.

It is the declaration generator a DB2 tool which generates DB2 equivalent COBOL variables called host variables.

DCLGEN member is used in the program as

```
EXEC SQL
    INCLUDE DECEMPPP
END=EXEC
```

23. DIFFERENCE BETWEEN INCLUDE & COPY.

Both INCLUDE & COPY statements will copy the statements from particular member to MAIN program after compilation.

Whenever DB2 is there, we will use INCLUDE.

COPY & INCLUDE will do same operation. But the difference is INCLUDE will be evaluated at the time of pre-compilation. Copy will be evaluated at the time of compilation.

24. WHAT IS CURSOR IN DB2.

To retrieve more than one row randomly from DB2 table, we are going to use the cursors.

STEPS TO USE CURSOR:

There are four steps to use cursor.

1. Declare cursor
2. Open cursor
3. Fetch cursor
4. Close cursor

NOTE:

-811 abend caused because the user trying to retrieve more than one row, without declaring cursor.

25. WHAT IS COMMIT.

It saves the changes permanently in the database.

26. IN WHAT WAY U WILL INSERT THE DATA INTO TABLE.

INSERT INTO TABLENAME VALUES (‘A001’, ‘RAHUL’, ‘ ‘)

27. WHAT IS BIND PARAMETER.

SYNTAX:

```
//SYSTSIN DD *
 DSN SYSTEM(DB9G)-
 BIND PACKAGE(PACKAGE-NAME) MEMBER(MEMBER-NAME)-
 OWNER(OWNER-NAME) QUALIFIER(QUALIFIER-NAME)-
 VALIDATE(BIND) ISOLATION(CS)-
 RELEASE(COMMIT) ACTION(REPLACE)-
 EXPLAIN(NO)-
 END
/*
//
```

SYSTEM: It tells which DB2 region, we are using.

PACKAGE: It specifies the package name into which the DBRM and best access path.

MEMBER: Represents COBOL-DB2 program member name.

OWNER: Owner of the table.

QUALIFIER: Used to unequally identify a test table when it has replication of tables.

1. Owner –ID and qualifier are same in production.
2. In development/testing- they are different.

VALIADTE: It checks the authorization of the user for binding or execution of the queries.

RELEASE: It releases locks on the table space.

1. COMMIT: Saves the changes made to the data in the database permanently.
2. ROLLBACK: It does not save the changes made to the database.

NOTE: 1. upon successful execution of the program, AUTO COMIT is issued.

2. upon un-successful execution of the program, AUTO ROLLBACK is issued.

ACTION: ADD/REPLACE

ADD: It's a new package, ADD will include the package into (SYSIBM.SYSPACKAGE)

REPLACE: It will override the existing package version with the new one.

EXPLAIN: EXPLAIN OF YES: Will give the best path, by using RUNSTAT utility after binding.

PACKAGE: DBRM+best access path

EXPLAIN OF NO: Will not give the best path after binding.

28. WHAT IS ISOLATION.

It specifies the level of locking. There are 3 isolation levels in DB2.

1. Cursor stability (CS). –row level locking
2. Repeatable read (RR). –page level locking
3. Uncommitable read (UR) –it wont lock any record.

29. HOW EXTRACTION OF SQL WILL DONE IN COBOL-DB2 APPLICATION.

SQL(Structered Query Language) is the language that is used to retrieve/update/delete DB2 data. SQL statements are embedded into COBOL program within the scope of 'EXEC SQL and END-EXEC.

COBOL-DB2 Program is first feed to DB2 pre-compiler that extracts the DB2 statements into DBRM and replace the source program DB2 statements with COBOL CALL statements. This modified source is passed to COBOL compiler and then link editor to generate load module. During pre-compilation, time stamp token is placed on modified source and DBRM.

30. IS DCLGEN MANDATORY.

Its optional in DB2. Without DCLGEN also, we can declare the variables in COBOL equivalent data types. But there will be a mismatch. To avoid those mismatch between the COBOL & DB2, we are generating equivalent data types using DCLGEN.

31. WHERE U WILL DECLARE A CURSOR.

Either in working-storage section or procedure division.

32. WHAT IS PRIMARY KEY & INDEX KEY.

Primary key:

- It is used to uniquely identify a row in the table.
- When primary key constraint set on a column it doesn't allow duplicate or null values.

INDEX KEY:

Will create the unique index. For the primary key field. It is useful to retrieve the data from DB2 table. (without unique index we cant create primary key).

33. WHAT UTILITY WE WILL USE IN COBOL-DB2 PROGRAMME.

For pre-compilation, the utility is "DSNHPC'. For BIND, the utility is "IKJEFT01".

34. WHAT OPERATIONS WE CAN SEE WHEN WE ARE COMPILING A COBOL-DB2 PROGRAMME.

Compilation of DB2 program involves

1) Pre-compilation

2) Bind

3) Execution

35. WHAT IS DCLGEN.HOW IT WILL GENERATE.

Its optional in DB2. Without DCLGEN also, we can declare the variables in COBOL equivalent data types. But there will be a mismatch. To avoid those mismatch between the COBOL & DB2, we are generating equivalent data types using DCLGEN.

36. WHAT IS COMMUNICATION AREA(CA).

SQLCA (structured query language communication area)

- It is the communication area between COBOL and DB2, it is used to know the status of a DB2 statement.
- It is used in the programming as

```
EXEC SQL
    INCLUDE SQLCA
END_EXEC
```

- This is resolved during pre-compilation and the below structure is copied.

```
01 SQLCA
   03 SQLAID PIC EX
   03 SQLCODE PIC
   03 SQLSTATE PIC
   03 SQLERR PIC
   03 SQLWARN PIC
```

- If SQLCODE=0

Successful execution & row found.

SQLCODE=+100

Successful execution & row not found.

SQLCODE=-VE

Unsuccessful execution.

37. WHAT ARE THE CONTENTS OF A DCLGEN.

It contains the DB2 column names and the data types, COBOL-field names,, another data type and data length.

38. IF WE HAVE 100 RECORDS,HOW WE RETREIVE EQUIVALENT RECORDS FROM TABLE USING SELECT.

By using Distinct we can retrieve the unique records from DB2 tables.

39. IN ONE TABLE ONE RECORD HAVE NULL VALUES.HOW CAN REPLACE THE NULL VALUES.

01 EMP-LANDLINE-NO PIC S9(4) COMP.

Whenever we want to pass value to null, that should be declared in WS section. Then only we can pass.

40. HOW CAN U DELETE THE DUPLICATES IN UR DATABASE BUT WE DON'T KNOW WHERE THE DUPLICATES ARE THERE.

- SELECT CLASS, SECTION, COUNT(*) FROM EMP_TBL
  GROUP BY CLASS, SECTION
  HAVING COUNT (*)>1

41. WHAT R STEPS IN CURSORS.

To retrieve more than one row randomly from DB2 table, we are going to use the cursors.

STEPS TO USE CURSOR:

There are four steps to use cursor.

1. Declare cursor
2. Open cursor
3. Fetch cursor
4. Close cursor

NOTE:

-811 abend caused because the user trying to retrieve more than one row, without declaring cursor.

42. HOW CAN U KNOW THAT TABLE CONTAINS NULL VALUES.

43. WHAT ARE THE PARAMETERS IN BIND.

SYNTAX:

```
//SYSTSIN DD *
 DSN SYSTEM(DB9G)-
 BIND PACKAGE(PACKAGE-NAME) MEMBER(MEMBER-NAME)-
 OWNER(OWNER-NAME) QUALIFIER(QUALIFIER-NAME)-
 VALIDATE(BIND) ISOLATION(CS)-
 RELEASE(COMMIT) ACTION(REPLACE)-
 EXPLAIN(NO)-
 END
/*
```

//

SYSTEM: It tells which DB2 region, we are using.

PACKAGE: It specifies the package name into which the DBRM and best access path.

MEMBER: Represents COBOL-DB2 program member name.

OWNER: Owner of the table.

QUALIFIER: Used to unequally identify a test table when it has replication of tables.

3. Owner –ID and qualifier are same in production.
4. In development/testing- they are different.

VALIADTE: It checks the authorization of the user for binding or execution of the queries.

RELEASE: It releases locks on the table space.

3. COMMIT: Saves the changes made to the data in the database permanently.
4. ROLLBACK: It does not save the changes made to the database.

NOTE: 1. upon successful execution of the program, AUTO COMIT is issued.

2. upon un-successful execution of the program, AUTO ROLLBACK is issued.

ACTION: ADD/REPLACE

ADD: It's a new package, ADD will include the package into (SYSIBM.SYSPACKAGE)

REPLACE: It will override the existing package version with the new one.

EXPLAIN: EXPLAIN OF YES: Will give the best path, by using RUNSTAT utility after binding.

PACKAGE: DBRM+best access path

EXPLAIN OF NO: Will not give the best path after binding.

44. WHAT IS MEANT BY ISOLATION LEVELS.

It specifies the level of locking. There are 3 isolation levels in DB2.

1. Cursor stability (CS). –row level locking
2. Repeatable read (RR). –page level locking
3. Uncommitable read (UR) –it wont lock any record.

45. WHAT IS CS,RR,UNCOMMITABLE READ.

1) **CURSOR STEBILITY(CS)**:- It is row level locking. An exclusive lock (X-Lock) is acquired on the row to be updated. The lock is released on the next updatable row.

ROW:- LOCK,UPDATE,RELEASE

2) **REPEATABLE READ(RR)**:- It is page level locking , An X-Lock is acquired on an entire page on which the row to be modified is present. The lock is released on the page only when the control moves to the next updatable page.
3) **UNCOMMITED READ(UR):-** It allows users to access the data which is modified but not yet committed. Here U-Lock is issued.

46. WHAT ARE THE ABENDS U WILL GET IN UR DAILY WORK IN DB2.

-805: REASON: DBRM/PACKAGE is not found in the PLAN.

-911: REASON: DEADLOCK. The program is used by some other.

SOLUTION: submit after sometime.

-922: REASON: AUTHOURIZATION FAILURE.

SOLUTION: Send mail to DBA to get access for particular tables. Or client will give the access to complet the JOB.

-818: REASON: TIMESTAMP MISMATCH**,** is a RUNTIME ERROR. Whenever there is a mismatch of timestamp between DBRMLIB & LOADLIB.

SOLUTION: In that case, we have to REBIND the program.

-811: REASON: the user trying to retrieve more than one row, without declaring cursor.

SOLUTION: Cursor has to be declared.

47. CAN I INSERT MORE THAN ONE RECORD AT A TIME INTO TABLE.

By using dynamic SQL, we can insert more than one row into the DB2 table.

In SPUFI, put ; at the end of each query, to execute more than one query and put '-' in front of the query, if you don't want to execute.

48. CAN I SELECT MORE THAN ONE TABLE AT A TIME.

SELECT * FROM TABLE1, TABLE2

49. WHAT IS SPUFI.

SQL processing using file input

50. I DIDN'T GIVE CLOSE CURSOR IN THAT CASE WHAT IT WILL HAPPEN.

If I CLOSE CURSOR is not declared, in that case we can't select the records from DB2 table.

51. WHAT IS RDBMS.

RDBMS-Relational Database Management System.

There is a relationship between the columns of DB2 table.

52. U KNOW ANY OTHER RDBMS.

No

53. CAN I SELECT MORE THAN ONE TABLE BY USING SELECT QUERY.

SELECT * FROM TABLE1, TABLE2

54. CAN I USE MORE THAN ONE PRIMARY KEY FOR A TABLE.

No I can't use more than one primary key.

55. WHAT IS THE TABLE SPACE WHAT IT CONTAINS.

Amount of space required to store the tables, views (simple objects). There are three types.

1. SIMPLE: In a page of TS, more than one table of data is stored.
2. SEGMENTED: In a page, single table data is stored
3. PARTITIONED: If a particular group of data is stored in a set of pages is called partitioned. Then it is called partitioned table space.

56. WHAT ARE THE ERROR CODES U KNOW IN DB2.

-805: REASON: DBRM/PACKAGE is not found in the PLAN.

-911: REASON: DEADLOCK. The program is used by some other.

SOLUTION: submit after sometime.

-922: REASON: AUTHOURIZATION FAILURE.

SOLUTION: Send mail to DBA to get access for particular tables. Or client will give the access to complet the JOB.

-818: REASON: TIMESTAMP MISMATCH**,** is a RUNTIME ERROR. Whenever there is a mismatch of timestamp between DBRMLIB & LOADLIB.

SOLUTION: In that case, we have to REBIND the program.

-811: REASON: the user trying to retrieve more than one row, without declaring cursor.

SOLUTION: Cursor has to be declared.

57. IN WHAT SITUATION WE WILL USE NULL INDICATORS.

If suppose I want to move null value to a land-line no. then that variable has to be declared in WORKING-STORAGE SECTION as follows.

01 variable-name PIC S9(4) COMP.

58. WHAT IS -805.

-805: REASON: DBRM/PACKAGE is not found in the PLAN.

59. IF I OPENED A CURSOR I DIDN'T CLOSE IT.WHAT WILL HAPPEN I DIDN'T GET ANY ERROR.

If we don't CLOSE CURSOR, that also will give abend. Because, if a table is opened in BATCH mode, then it can't be opened in CICS region, (i.e., ONLINE mode), if we don't close the cursor.

60. WHAT IS THE UTILITY FOR RUNNING A COBOL-DB2 PROGRAMME.

For pre-compilation, the utility is "DSNHPC'. For BIND, the utility is "IKJEFT01".

61. WHAT IS EXPLAIN IN DB2.

EXPLAIN OF YES: Will give the best path, by using RUNSTAT utility after binding.

PACKAGE: DBRM+best access path

EXPLAIN OF NO: Will not give the best path after binding.

62. WHAT IS THE SYNTAX OF DECLARE CURSOR.

Declaring the cursor :-

```
EXEC SQL
    DECLARE CURSORNAME CURSOR FOR
        SELECT * FROM TABLENAME
        END-EXEC.
```

Declare cursor statement just creates the cursor declaration can be given 'working storage selection' or ' procedure division '.

To keep the cursor open even after 'commit' is issued.

```
EXEC SQL
    DECLARE CURSORNAME CURSOR WITH HOLD
    FOR SELECT * FROM TABLENAME
END-EXEC.
```

NOTE:-

It I want to "commit" the updations ,(we are doing for the table) and again I want to update next 10 records, at that time the cursor will automatically closed because of "COMMIT".If we want to hold the cursor until all the records are updated. Then cursor should declare with "WITHHOLD" option.

63. WHAT IS -818 ABEND.ANYTIME U COME ACROSS THIS.HOW U WILL OVERCOME THIS.

TIMESTAMP MISMATCH, -818, is a RUNTIME ERROR. Whenever there is a mismatch of timestamp between DBRMLIB & LOADLIB. In that case, we have to REBIND the program.

64. WHAT IS -811.

To retrieve more than one row randomly from DB2 table, we are going to use the cursors.

STEPS TO USE CURSOR:

There are four steps to use cursor.

1. Declare cursor
2. Open cursor
3. Fetch cursor
4. Close cursor

-811 ABEND CODE caused because the user trying to retrieve more than one row, without declaring cursor.

65. WHAT IS THE UTILITY FOR RUN.

IKJEFT01

66. IN BINDING WHAT IT WILL CREATE EITHER PLAN OR PACKAGE.

Binding process can happen in two stages, BIND PACKAGE and BIND PLAN. One DBRM is created for one program. If the main program calls n number of sub-programs, then there will be n DBRMS in addition to main program DBRM. These n+1 DBRM can be directly feed to BIND PLAN to produce a single PLAN or create m number of intermediate packages each formed by one or more DBRM. This m numbers of packages are then feed to BIND PLAN step that produces PLAN. Package is not executable but PLAN is executable. To run a DB2 program, PLAN is mandatory.

67. WHAT ARE STEPS U HAVE TO FOLLOW AFTER COMPLETION OF WRITING COBOL-DB2 PROGRAM.

After completion of writing COBOL-DB2 program, compilation will be done which involves,

1) Pre-compilation

2) Bind

3) Execution

68. WHAT WILL HAPPEN IN PRECOMPILATION.

COBOL-DB2 Program is first feed to DB2 pre-compiler that extracts the DB2 statements into DBRM and replace the source program DB2 statements with COBOL CALL statements. This modified source is passed to COBOL compiler and then link editor to generate load module. During pre-compilation, time stamp token is placed on modified source and DBRM.

69. WHAT U WILL GIVE IN RUN PGM.

DB2 region name, program to run, DB2 plan and PARM(if needed) are provided.

i-One Techsolutions

CICS Interview questions

1. How will you run a online program. ?

By using Transaction ID, online program can run.

2. How will you create maps in cics ?

Two ways.

A. BMS MACROS.
B. SDF TOOL (SCREEN DEFINITION FACILITY)
   DFHMSD – Mapset
   DFHMDI – Map
   DFHMDF - Field

3. What are the types of maps?

Two types of MAP.

1. **Physical Map:** controls the screen alignment, sending and reccieving of constants and data to and from a terminal. They are coded using BMS macros, assembled and link edited into CICS LOAD LIBRARY. They ensure the device independence in application programs.
2. **Symbolic Map:** defines the map fields used to store the variable data referenced in COBOL program. They are also coded BMS macros. But after assembling, they are placed in a copy library and then copied into CICS programs using COPY statements. They ensure device and format independence to the application program.

4. What is MDT.

Modified data tag

MDT is one bit field of the attribute byte. The program can receive only the fields with MDT '1' on RECEIVE. Effective use of MDT can reduce the data traffic drastically in the communication line.

MDT can be SET/RESET in the following ways:

1. When the user modifies the field, the MDT of the field is automatically set to ON.
2. CTRL=FRSET of DFHMSD or DFHMDI will RESET the MDT to 'OFF' for all the fields In the mapset or map. FSET keyword of the attribute operand definition of DFHMDF will set the MDT to 'ON'. It overrides the FRSET definition for the specific field.
3. Before sending the screen, by overriding the MDT bit of attribute byte of the field the MDT can be set to 'on'.

5. What are the parameters in DFHMSD

DFHMSD (defined field hierarchy map set definition)

Mapsetname DFHMSD TYPE=&SYSPARM/MAP/SECT,

MODE=INPUT/IN/OUT,

T10APEX=YES,

LANG=COBOL/ASM/PL,

STORAGE=AUTO,

CTRL= (FREEKB,FRSET)

1. TYPE: If specifies the type of the map to be generated.
   &SYSPROM: It will generates both physical and symbolic map
   MAP: It generates only the physical map
   DSET: It will generate only symbolic map
- Physical map is the load module of the BMS MACROS
- Symbolic map is the copy book generated for the fields with field name.

2. MODE: It specifies the type of use of map.
   INPUT: It enables the map to be used for both Input and output pupose.
   IN: for input maps lik entry screens
   OUT: for output maps like display screens
3. TIOAPFX (terminal input output area prefix): when TFOAPFX=YES then system generates a 12 byte filler in which the control information pressed by the user.
   The 12 byte memory is TIOA
4. LANG: It specifies the programing language with which we can use the map.
5. STORAGE: When STORAGE=AUTO, is specified system allocates a separate memory for the map set.

6. Difference between XCTL&LINK

XCTL is used to pass control from one program to another but doesn't expect the control to return back.

SYNTAX:

EXEC CICS

XCTL PROGRAM(PROGRAM NAME)

END EXEC

LINK: It is used to pass control from one prog. to another but the control is expected to return back.

7. Difference between conversation&psuedo conversation

CONVERSATION: processing transaction is called conversation.

PSUEDO CONVERSATION: for sending and receiving of message, we will use pseudo conversation.

8. Syntax to use maps in cobol -cics program

COPY MAPSETNAME**I**, MAPSETNAME**O**

9. What is EIBCALEN

EXECUTE INTERFACE BLOCK COMMUNICATION AREA LENGTH

If EIBCALEN = 0

Indicates user not entered any data

ELSE

User entered some data

SYNTAX:

```
IF EIBCALEN = 0
        PERFORM SEND-PARA
        ELSE
        PERFORM PROCESS-PARA
```

10. What is DFHCOMMAREA

It is the communication area between the two programs or two tasks. COMM area is declared in the COBOL program as follows.

```
SYNTAX:     LINKAGE SECTION
                01  FHCOMMAREA PIC X(100)
```

In every CICS program DFHCOMMAREA is declared and the maximum data length can be 32767(32k). It is used to know whether he user has entered data in the screen or not It stores the count of characters entered on the screen. If EIBCALEN is '0' then it indicates user has not entered any data. ELSE User has entered data.

Note: If NO DFHCOMMAREA declared then system takes a one byte (1B) DFHCOMMAREA.

11. DFHRESP options

```
WHEN DFHRESP (NORMAL)      -> FILE IS FOUND
WHEN DFHRESP (NOT FOUND)-> FILE IS NOT FOUND
WHEN DFHRESP (LENERR)      -> LENGTH ERROR
WHEN DFHRESP (DUPREC)      -> DUPLICATE RECORD
```

12. How to use VSAM file in COBOL- CICS program

Reading the records sequentially in CICS:

```
EXEC CICS
        START BR FILE(FILE-NAME)
        RIDFLD(POL-NUM)                    ->record identification field
        KEYLEN(LENTH)
END-EXEC.
EXEC CICS
        READ NEXT FILE(LOG.FILE-NAME)
        RIDFLD(FIELDNAME)
END-EXEC.
```

13. TYPES OF CURSORS IN CICS

1.STATIC CURSOR POSITIONING: USING IC

2.DYNAMIC CURSOR POSITIONING:

Dynamically to place the cursor into a desired field we need to move '-1' to its length.

MOVE 'PLEASE ENTER CORRECT PASSWORD' TO ERRMSG0

MOVE -1 TO PWDL

PERFORM 1600_SEND.

3.RELATIVE CURSOR POSITIONING:

Relatively the cursor is positioned at a desired field by specifying the relative byte position compared to the first byte.

```
EXEC CICS
    SEND MAP('MAP1')
         MAP('MAPST1')
         CURSOR(842)
END_EXEC.
```

14. How will you create COBOL+CICS+DB2 program

15. What are the commands to execute MAP& PROGRAM

CEDA-CICS execution, definition and administration. Command is used to define and install mapset of PGM.

By using CECI(CICS execution command interruption) command is used to send map to online screen.

16. What are the types of QUEUEs

Queues are of 2 types:

1. TSQ (temporary storage queue)
2. TDQ (transient data queue)

17. TSQ&TDQ

1. TSQ: Queue records are written based on item numbers. TSQ records can be accessed sequentially, dynamically or randomly.
2. TDQ: records are written without item number. Records can be read only sequentially. A TDQ record once read is deleted automatically.

18. In which QUEUE we can implement PAGE UP&PAGE DOWN

In TSQ

IMS-DB Interview Questions:

1.WHAT IS RESTART PARAMETER IN IMS.

RESTART parameter is used to restart the execution from last appended record

2.IF I HAVE 1 MILLION RECORDS I START UPDATING AFTER UPDATING 50,000 RECORDS I CANCELLED THE JOB.I WANT TO START EXECUTION FROM BEGINING IN THAT CASE WHAT I HAVE TO DO.

CHECK POINT=NORMAL in PARM parameter

3.WHAT ARE PCB'S,PSB'S,DBD.

DBD- Physical Layout of the Database.

PSB- Programmatic view of the Database.

PCB- Logical view of the Database.

4.WHAT ARE THE MODES OF PROCESSING.

There are three types of modes of processing.

1. BATCH DL/I
2. MPP (MESSAGE PROCESSING PROGRAM)
3. BMP (BATCH MESSAGE PROCESSING)

5.HOW MANY CALLS U USED IN IMS-DB.

Two types.

1. Entry statement
2. Call statement.

6.HOW DO U CALL AN IMS-DB PROGRAM.

CALL 'CBLTDLI' USING function code, PCB-Mask, I-O area, SSA-1, SSA-2,..

7.HOW CAN U VIEW A IMS-DB DATABASE.

FOR FILEAID

STEP1: TSO FILE MENU

STEP2: BROWSE/VIEW/EDIT

STEP3: DBD NAME:-

DBD Library:-

PSB Library:-

REGION NAME:-

i-One Techsolutions

STEP4: It displays the segments in the database.

A ,…

B,..

C,..

NOTE: to know the key value, type K infront of the Segment name,. it will display the KEY value..

8.WHERE THE PHYSICAL DATABASE EXIST IN IMS-DB.

In DBD(database description)

9.WHAT IS WHERE CONDITION EQUIVALENT IN IMS-DB.

SSA(segment search argument)

10.WHAT IS THE I-O AREA CONTAINS.

It contains the standard record description, an area in working storage section, which holds the segments for manipulation. Used to pass segments between PGM to PGM.

11.HOW CAN U ACCESS IMS-DB.

Using ACCESS parameter in DBD definition like HS, HD.

12.WHAT DBD CONTAINS.

DBD name and ACCESS method

13.WHAT IS PCB&PSB.

PSB: describes the programmatic view of the database and access authority to the application program in the database.

PCB: describes the logical view of the database.

14.IN WHICH ORDER WE WILL DECLARE PCB'S IN ENTRY STATEMENT.

In sequential order.

15.WHAT IS DL/I.

It is an interface between application program and IMS-Database

16.WHENEVER WE ARE UPDATING A DATABASE IN THAT SITUATION WHICH FUNCTION CODE WE WILL USE.

GHU,GHN,GHNP

17.WHAT IS DBD.

It is the physical layout of the database. DBA defines the physical layout using DBDGEN macro

18.WRITE THE DBDGEN MACRO.

PRINT NOGEN

```
DBD         NAME=VENDOR,ACCESS=HDAM

DATASET     DD1=VEND,DEVICE=3380

SEGM        NAME=VENSEG,PARENT=0,BYTES=10

FIELD       NAME=(VENCODE,SEQ,U),BYTES=10,START=1,TYPE=C

SEGM        NAME=ITEMSEG,PARENT=VENSEG,BYTES=5

FIELD       NAME=(ITEMCODE,SEQ,U),BYTES=5,START=1,TYPE=C

SEGM        NAME=LOCNSEG,PARENT=ITEMSEG,BYTES=9

FIELD       NAME=(LOCNCODE,SEQ),BYTES=3,START=1,TYPE=C

FIELD       NAME=ORDDATE,BYTES=6,START=4,TYPE=C

DBDGEN

FINISH

END
```

19.IN UR COBOL-IMSDB PROGRAMME WHERE U WILL GIVE PSB NAME.

In RUN JCL, PARM parameter use PSB-name in COBOL-IMS-DB.

20.AFTER DOING SOME OPERATION ON SEGMENTS WHERE THE OUTPT HAS TO BE DIRECTED.

To I-O Area

21.WHAT IS SSA

- The SSA is an area in working storage used to identify the segment to which access is required
- It is an optional dl/i call parameter that qualifies a call by specifiying a particular segment type or occurrence

When specified in a dl/i call

- The SSA should always follows I-O parameter
- There may be 1 to 15 SSA in a call statement
- The SSA must appear in hierarchical order by segment type
- The SSA contains information to obtain the required segment. The information could by segment name or segment name & segment key or segment name & search field

22.WHAT IS RUN-JCL FOR COBOL-IMSDB PROGRAMME.

```
//JOB CARD

//STEP1 EXEC PGM=DFSRRC00

            PARM=(PSBNAME,PGMNAME,MODE OF ACCESSING,CHKPT=NORMAL)
```

23.WHERE U WILL GIVE PROGRAMME NAME IN RUN-JCL.

IN PARM PARAMETER.

24.WHAT IS BMP MODE.BY SEEING THE JCL HOW CAN U KNOW THAT IT IS WHICH TYPE OF ACCESS.

BMP: Batch message processing. There are two types.

1.Transaction Oriented:

Can access files and database online. Read and write online message queues.

2.Batch Oriented:

Read message queues. Can access online database in batch mode.

By seeing the PARM parameter of the RUN JCL, we can identify the mode of access.

25.WRITE THE GENERAL CALL STATEMENT.

CALL 'CBLTDLI' USING function code, PCB-Mask, I-O area, SSA-1, SSA-2,..

26.WHAT ARE THE ACCESS METHODS AVAILABLE.

There are six types.

1. HSAM(HIERARCHICAL SEQUENTIAL ACCESS METHOD)
2. HISAM(HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD)
3. SHSAM(SIMPLE HIERARCHICAL SEQUENTIAL ACCESS METHOD)
4. SHISAM(SIMPLE HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD)
5. HDAM(HIERARCHICAL DIRECT ACCESS METHOD)
6. HIDAM(HIERARCHICAL INDEXED DIRECT ACCESS METHOD)

27.WHAT IS THE DIFFERENCE BETWEEN HSAM &HDAM.

HSAM: Efficient for sequential accessing.

HDAM: Efficient for random accessing

28.I WANT TO RETRIEVE FROM 4TH SEGMENT ONWARDS HOW U WILL DO IT BY HSAM.

Its not possible in HSAM, because HSAM is only for sequential processing. Using HDAM, it is possible.

29.WHAT IS I-O AREA.

It contains the standard record description, an area in working storage section, which holds the segments for manipulation. Used to pass segments between PGM to PGM.

30.WHAT IS PCB-MASK CONTAINS.

There should be atleast one PCB for each database used. Status code of PCB mask explains whether a call is successful or not. The sequence in which PCB masks are defined in linkage section need not be same as that in PSBGEN macro

i-One Techsolutions